

UNCLASSIFIED

AD 295 651

*Reproduced
by the*

**ARMED SERVICES TECHNICAL INFORMATION AGENCY
ARLINGTON HALL STATION
ARLINGTON 12, VIRGINIA**



UNCLASSIFIED

NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

295651

295651

(92)

MEMORANDUM
RM-3339-PR
DECEMBER 1962

CATALOGED BY ASTIA
AS AD NO. 1

PRELIMINARY CODES AND RULES FOR THE AUTOMATIC PARSING OF ENGLISH

Jane J. Robinson



PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

MEMORANDUM

RM-3339-PR

DECEMBER 1962

**PRELIMINARY CODES AND RULES FOR
THE AUTOMATIC PARSING OF ENGLISH**

Jane J. Robinson

This research is sponsored by the United States Air Force under Project RAND - Contract No. AF 49(638)-700 - monitored by the Directorate of Development Planning, Deputy Chief of Staff, Research and Technology, Hq USAF. Views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of the United States Air Force. Permission to quote from or reproduce portions of this Memorandum must be obtained from The RAND Corporation.

The RAND Corporation

1700 MAIN ST • SANTA MONICA • CALIFORNIA

PREFACE

This Memorandum is an interim report on automatic machine analysis (parsing) of English which will eventually enable the electronic computer to receive natural English sentences and convert them internally into formal structures suitable for further machine manipulation.

In this Memorandum the bases upon which a machine will accept and act upon English sentences are given, together with the codes and rules, and the approach to be used in the further development of parsing is analyzed.

The immediate purpose is to develop a parsing program for automatic information retrieval. Work in information retrieval now goes beyond statistical, probabilistic methods, and as probes for relevant answers to requests have been refined by adding information about sentence structure, the need for something more than ad hoc or "quick and dirty" methods of sentence structure determination is increasingly apparent.

The development of machine parsing has been undertaken by The RAND Corporation as a part of its continuing research effort in the field of computer technology as an adjunct to further studies in information retrieval, Command and Control, etc.

The approach to automatic parsing was established through discussions with Charles F. Hockett, Professor of

Linguistics and Anthropology, Cornell University, and Consultant to The RAND Corporation, and David G. Hays of RAND. The first program to test the codes and rules on a digital computer was designed by Robert Dupchak, Consultant to The RAND Corporation.

SUMMARY

This Memorandum presents a set of grammar codes and rules for analyzing, or "parsing," English sentences automatically on a digital computer. The introduction briefly discusses the nature of parsing and the general model of English structure on which this approach is based, and the following sections explain and illustrate the codes and rules in detail.

Although the computer program is not explained at length, the logic underlying it is illustrated in the discussion of the rules. A complete list of the rules written so far and a sample output of parsed sentences are appended.

The linguist and the programmer will find the presentation sufficiently detailed so that they can add to, refine, or modify the codes and rules and design programs for applying them to text.

CONTENTS

PREFACE	iii
SUMMARY	v
TABLES	ix
Section	
I. INTRODUCTION	1
Parsing as Interpretation	1
A Model for a Machine Grammar of English ..	3
II. THE GRAMMAR CODES IN THE GLOSSARY	6
The Problem of Homography and Ambiguity ..	6
Glossary Code Format	7
The Major Classes	9
Particles, Relatives, and Interrogatives	11
The Content-Word Classes	15
The Determiners	22
To, Be, Auxiliaries, and Contracted Forms	27
Odd Forms, Numerals, Foreign Words	30
III. THE PARSING RULES	32
Method of Application	32
Rule Symbols	34
Cover Symbols	35
Instruction Symbols	36
Rule Format	39
IV. RESULTANT CODES	42
Types	42
Multiple Resultants	50
V. THE OUTPUT	51
Appendix	
A. RULE SYMBOLS	57
B. RULES	59
C. SAMPLES OF PARSED SENTENCES	71

TABLES

Table

1. Major Parts-of-Speech Classification:	
Position 1 Codes	10
2. Unique Particles	11
3. Prepositions, Adverbs, Conjunctions	13
4. Relatives and Interrogatives	14
5. Verb, Noun, Adjective Codes	16
6. Determiner Codes	25
7. Coded Determiners	26
8. To, Be, Auxiliaries, Contracted Forms	28
9. New Type C Resultant Code Symbols	45

I. INTRODUCTION

PARSING AS INTERPRETATION

Interpretation of the expression these yellow clothes by an English speaker involves his experience with language and also with the non-linguistic world. That is, he uses his past experience to make distinctions between the colors yellow, red, and blue and the objects clothes, rags, and so forth, as well as between words that refer to them. So far as the words are concerned, machines can make distinctions too. The word yellow is different from the word red in certain definitely specifiable physical ways and it is not difficult to build reactions to such differences into machinery. What a machine cannot do, so far at least, is determine whether in a given context the expression is appropriate, whether or not there are objects in its experience that are appropriately called clothes and appropriately called yellow and appropriately referred to by these rather than those. However, insofar as the meaning of an utterance is a function of the physical contrasts among the shapes of sequences of words, a machine can be instructed to differentiate between these yellow clothes, these blue clothes, those red rags, and the quick brown fox jumped over the lazy dog. In a very superficial sense the machine's reaction to these contrasts can be considered an interpretation or even a translation.

But there is a different kind of contrast in language which must also be interpreted. In the expression, "These yellow clothes and these whiten them," the first three words contrast in one way with the last three and in a different way with the identically shaped sequence appearing in, "I like these yellow clothes." To instruct a machine to interpret the second kind of difference is to teach it to parse. Automatic digital computers are capable of receiving such instruction, but first, grammars must be written for them.

The English codes and parsing rules being developed at RAND are essentially a machine grammar. So far, this grammar has enabled the computer to parse successfully a variety of sentences which include coordinate, subordinate, relative, indicative, and interrogative clause structures. It will be further evaluated by having the computer apply it to the parsing of large amounts of unedited text. It is obviously incomplete and inadequate at present, even more so than grammars written for people, but it can be easily expanded and revised. It represents a fruitful approach to the problem because it takes advantage of the computer's ability to perform a large number of simple operations with great speed, it can be written very compactly, and it can be programmed by exact or algorithmic rather than by empirical or heuristic methods. Moreover, it does not force a

single interpretation on a sequence of words but is designed to preserve ambiguities that actually exist, so that loss of information in the course of parsing is reduced.

A MODEL FOR A MACHINE GRAMMAR OF ENGLISH

The relationships among words in meaningful sequences¹ are structured, and grammars can be viewed as models of the structures in the language for which they are devised; or at least as based on or providing insight into a model underlying those structures. In some languages the shapes of the words with their derivational and inflectional affixes play a major part in providing information for establishing their relationships with each other. In English and some other languages, word order is crucial though not decisive. How one gets from the linear or temporal ordering of elements to their structural relationship is, then, a major task to which a grammatical analysis of English sentences must address itself. It appears possible in English to consider the words in a sentence as projections from the branchings of a tree-like structure. Above the word-level, the arrangement of the branches appears to be binary in general so that

¹ It is customary to regard the sentence as the largest grammatically structured sequence in a language; the relationships among sentences in paragraphs and larger units are relegated to problems of style.

each node tends to have not more than two branches. We are accustomed to descriptions of sentence structure that are basically binary: a sentence consists of subject and predicate; the subject consists of a substantive and its modifiers; the predicate consists of verb phrase and complements; the verb phrase consists of verb and auxiliaries, and so forth. The elements of each substructure are usually adjacent and any departure from this arrangement in English may be very significant, as in the inversion of subject and auxiliary to differentiate statements from questions.

One can easily think of examples for which this analysis is unsatisfactory, but it does provide a model for powerful generalizations about English sentences and it lends itself to machine manipulation. For these reasons, the parsing program is based on this "immediate constituent" analysis. A computer is programmed to take sentences as input and to relate the words in them in paired substructures or "constitutes" until all of the material in a sentence has been accounted for. The output is a description of the structural relationships of the words in each sentence which may subsequently be used as input for information retrieval, machine translation, content analysis, linguistic research, or even as machine instructions.

In the process, the computer is supplied with two kinds of information: information about the potential each word has for entering into grammatical constructions with other words (forming constituents) and information about the environments in which these potentials are realized. The first is contained in a glossary which attaches to each word or form occurring in a text a code indicating the syntactic role or roles it may play. This is its parts-of-speech category. The second is contained in a set of parsing rules which, theoretically, lists all of the grammatically permissible combinations of adjacent codes, and supplies a new code called a resultant for each combination. These are the two parts of the machine grammar which will be presented in detail.

II. THE GRAMMAR CODES IN THE GLOSSARY

THE PROBLEM OF HOMOGRAPHY AND AMBIGUITY

English abounds in homographs, words whose shapes are identical but whose meanings and syntactic properties are widely different, e.g., bear ("ursus," noun) and bear ("sustain," verb). Homography of verb and noun is especially prevalent, e.g., walk, run, set; and all three of the main word classes--verb, noun, and adjective--exhibit it to a degree likely to surprise anyone who has not scanned word lists with a view to assigning parts-of-speech categories. This homography accounts for the ambiguity in the sequence these yellow clothes, cited previously. Often the addition of a single item of context is sufficient to resolve the ambiguity (cf., the bear and to bear); sometimes a complete sentence lends itself to two or more syntactic interpretations even when the total context is considered. This problem of ambiguity in words and constituents can be met, in part, by assigning multiple codes to each homograph or ambiguous sequence; that is, each possible classification could be coded separately. Another way of handling the problem is to devise a system of classification that recognizes recurrent patterns of homography and ambiguity. The latter method is the one adopted here. Words that can occur

only as verbs are coded with a 1 in first position, words that occur only as nouns with a 2 in first position, but those that can be either are coded with a 3. This principle of classification is followed when patterns of homography and ambiguity are sufficiently general. Only peculiar homographs such as well, might, can, and that are given more than one code. Well, for example, is given one code as verb/noun and another as an adverbial adjective, since this is an unusual combination of syntactic ranges. That is assigned the same code as this in the determiner group, but also receives a unique code as a relative particle. All forms of have and the finite forms of do have separate codes as verbs and as auxiliaries. This method of coding is geared to a parsing logic developed by John Cocke of IBM, which preserves ambiguities until they are resolved by the addition of larger context.

GLOSSARY CODE FORMAT

Codes have been attached to words in a glossary consisting of approximately 7000 entries compiled from text. Each word is keypunched in columns 1-30 of a standard IBM card. Columns 35-41 contain a word-number; columns 49-52 contain the code. Forms which are simply inflected variants of a common stem are assigned the same word-number in order

to reduce the duplication of semantic information that may be stored with them later; however, peculiar homographs are entered twice, with separate word-numbers for each entry. For example the series do, does, did, doing, done has one word-number as forms of the main verb and do, does, did has another word-number as forms of the auxiliary. So far, no need has arisen for assigning more than two word-numbers to a form.

The code attached to each word or form consists of a string of four digits. The digit in the first position indicates the major parts-of-speech assignment, those in second and third positions subclassify by coding the suffixes of inflected words or the syntactic properties of uninflected ones. Function words with unusual or unique syntactic properties and forms containing numerals and mathematical symbols are given special codes. Digits in the fourth position code some of the verb and noun subclasses of individual words and the classes of constituents established during parsing.

The most regular part of the system is the verb-noun-adjective-determiner-auxiliary section running from 1000-9500. These groups have morpho-syntactic properties of tense, number, or degree, which furnish general criteria

for classification. In contrast with these, the forms coded with a 0 in first position are a mixed lot, mostly uninflected, many with unique functions. It has seemed best at this stage to isolate several words with unusual properties and assign them arbitrary codes at the beginning of the 0xxx series; consequently the "system" here is a somewhat disorderly compromise between classification and partial listing. No attempt was made to provide codes for all of the uninflected particles but only for those actually appearing in our limited text.

At the other end of the system, codes 96xx have been temporarily reserved for numerals, alphabet sequences not known to be acronyms, mixtures of numerals and letters, and foreign words. It is assumed that the numerals and alphabet sequences will function like nouns and noun modifiers for the most part, but until further investigation, we prefer to keep them separate from the main word classes and particles.

THE MAJOR CLASSES

Table 1 illustrates the major classes distinguished by the first position codes 0-9.

Table 1
Major Parts-of-Speech Classification: Position 1 Codes

Code	Meaning	Examples
0	Unique forms; uninflected particles (adverbs, prepositions and conjunctions); relatives and interrogative pronouns. (Note: adverbs formed by adding <u>-ly</u> to an adjective base are coded as inflected adjectives.)	very, and, why, in, by, that, who
1	Verb only.	see, civilize
2	Noun or pronoun (excepting pronouns that may also be determiners, e.g., <u>his</u>).	death, civilization, John, he, theirs
3	Verb or noun.	dance, telephone
4	Adjective only (including adverbs with <u>-ly</u>).	true, beautiful
5	Verb or adjective.	clean, clear
6	Noun or adjective.	kind, corporal
7	Verb, noun, or adjective.	brief, yellow
8	Determiners (articles, demonstratives, quantitatives, possessive pronouns).	a, the, this, many, five, my
9	<u>To</u> , <u>be</u> , auxiliaries.	to, be, have, do, can

Particles, Relatives, and Interrogatives

Subclassification and listing of the words coded 0 in first position are illustrated in Tables 2, 3, and 4. Only three of the four positions allotted in a complete code are used, so all fourth position codes are zero. This position will be used for further subclassification after the program has been tested to see what other adjustments are needed.

Table 2

Unique Particles

Code	Words	Comments
0010	and, and-or	Coordinating conjunctions appearing in the working text.
0020	or	
0030	but	
0040	not	These vary in syntactic function,
0050	only	but all can modify adjectives and
0060	very	some types of adverbs.
0070	as	
0080	than	
0090	too	
0100	somewhat	
0110	how	These can be both adverbs, conjunc-
0120	when, where	tions, and interrogatives. As inter-
0130	why	rogatives, they appear with inverted
0140		order of subject and auxiliary.
0150	there	These pro-adverbs are separated from
0160	here	the rest because of their use in
		inversions. "There" is a common
		inverter.
0190	etc.	Will require special handling when
		rules are written for punctuation.

Table 2 lists specific words with their arbitrary codes, occupying a block of numbers running from 0000-0190. They are grouped very roughly on the basis of some shared syntactic characteristics, noted in the comments.

Table 3 illustrates the subclassification of prepositions, adverbs, and conjunctions. The principle of coding to indicate homography and ambiguity is applied, so that a word like after, which functions as preposition, adverb, or conjunction, is assigned to a class whose code symbol in second position differs from symbols assigned to with, again, or if, which are all members of unambiguous classes. Not counting the null class, seven classes are theoretically possible, but the category "adverb/conjunction" has been omitted because the only words belonging to it are how, when, where, and why, which have already received arbitrary codes.

This group, occupying the block of codes 02xx-07xx, is further subclassified in third position on the basis of ability to postmodify a noun or premodify a simple, unaffixed verb. At the level of individual words, this subclassifies only adverbs, which by definition can all modify verbs. Words that are only prepositions or only conjunctions perform none of these functions, although constituents containing them can.

Table 3
Prepositions, Adverbs, Conjunctions.

Pos. 1		Position 2		Position 3		Pos. 4	
Code	Code	Meaning	Example	Code	Meaning	Example	Code
0	2	Preposition	with	0	Can postmodify a noun	abroad	0
	3	Adverb	soon	1	Can premodify a verb	soon	
	4	Conjunction	if				
	5	Prep/adv	up	2	Can do either	also	
	6	Prep/conj	for	3	Can do neither	with, if	
	7	Prep/adv/conj	after				

Table 4
Relatives and Interrogatives

Code	Word	Description
0800	that	relative only, singular/plural agreement
0810	what	interrogative, determiner, singular/plural agreement
0820	which	relative, interrogative, determiner, singular/plural agreement
0830	who	relative, interrogative, singular/plural agreement, nominative
0840	whose	relative, interrogative, determiner, singular/plural agreement
0850	whom	relative, interrogative, singular/plural agreement, accusative
0860	how much	interrogative, determiner, singular agreement
0870	how many	interrogative, determiner, plural agreement

Consequently, the glossary codes for adverb classes and subclasses will also be resultant codes for constituents containing the other two classes.

Table 4 lists the codes for a group of complexly related words which share certain functions as relatives, interrogatives, and interrogative determiners, although only two of them combine all of these functions. In this group, that is only relative. Another that, with a separate word number, also appears in the determiner group (8xxx) where its code is identical with the code of this, since as determiners, this and that are symmetrical with these and those, and general rules can be written for their parsing. Table 4 includes arbitrary codes for the constituents how much and how many.

The Content-Word Classes

Table 5 contains the codes for subclasses of the content-word classes with the first position codes repeated for easier reference. The reader will note that two positions are used to code the inflectional suffixes. The order of positions in which the suffixes are coded is conditioned by the peculiarities of English homography. As noted previously, English nouns and verbs are frequently homographic. The plural suffix of nouns is also homographic with one of

Table 5
Verb, Noun, Adjective Codes

Position 1		Position 2			Position 3			Position 4		
Code	Meaning	Code	Meaning	Example	Code	Meaning	Example	Code	Meaning	Example
1	V	0	No suf- fix	brief	0	No suf- fix	took	0	Non-determined N or A	death simple
2	N	1	Past	took	1	Verb <u>s</u>	takes	1	Intransitive V, non-determined N or A	walk long
3	V/N	2	Present parti- ciple	dancing	2	N plural	boys men	2	Transitive V; Non-determined N or A	brief
4	A	3	Past parti- ciple	taken	3	V/N <u>s</u>	dances	3	Two-object V; Non-determined N or A	give make light
5	V/A	4	Past/ past parti- ciple	briefed	4	N sing- ular or plural	fish	4	Determined N	John's
6	N/A	5	Present tense/ past parti- ciple	run	5	A <u>ly</u>	truly	5	Completely determined N	John he, him
7	V/N/A	6	Present tense/ past/ past parti- ciple	set	6	N accu- sative	him			
		7	Compar- ative <u>er</u>	nicer blacker						
		8	Superla- tive <u>est</u>	nicest						
		9	Genitive <u>'s</u>	man's						

the verb suffixes, so that dances, for example, may be either a plural noun or a present tense verb requiring a singular, third-person subject. But homography is also present within the verb inflections themselves. Set is not only ambiguously a noun or a verb, as a verb it is ambiguously an infinitive, a present tense form requiring a plural subject, a past tense, or a past participle. Compare:

The set of all objects. The sets of all objects.

They set it. He sets it.

and, They wanted to set the They wanted to go there.
table.

They set the table every They go there every day.
day.

They set it yesterday. They went there yesterday.

They have set it. They have gone there.

The homography within the verb form is coded in second position; the remaining homography of verb with noun is then coded in third position.

Verbs and nouns are subclassified in third position according to the presence or absence of the -s suffix. A zero means that it is absent, a 1 that it is present in an unambiguous verb (10lx), a 2 that it is present in an unambiguous noun (2x2x), a 3 that it is present but the stem is ambiguously noun or verb (3x3x); a 4 that it is absent

but that as a noun, the form is ambiguously singular or plural. The code symbol 6 is assigned to accusative forms of nouns (pronouns), regardless of whether or not they are singular or plural, since they do not appear as subjects nor take determiners, and therefore do not require number agreement so far as parsing is concerned.

Verbs are subclassified in fourth position according to their ability to take one or more substantive objects. An intransitive verb has a 1 in fourth position and the parsing rules do not provide for its being parsed with a following substantive expression. A 2 in this position indicates that the verb is transitive, a 3 that it can take both an indirect and a direct object. How these codes are modified during parsing is discussed in the section on Resultant Codes (Sec. IV). Here it is sufficient to note that all potential verbs will have a 1, 2, or 3 in fourth position.

Nouns are subclassified in fourth position according to their ability to pick up determiners. Boys, for example, may form a constitute with the or all or both together, while John's may form a constitute with all, but not with the, and John and he never take determiners. The code for boys is 2020, for John's 2904, and for John and he 2005. Unambiguous

nouns will have a 0, 4, or 5 in fourth position. Words in the ambiguous V/N or V/N/A classes will have a 1, 2, or 3 in fourth position, depending on the subclass of the verb, but as nouns they are capable of acquiring determiners and the codes 1, 2, 3 are equivalent to 0 for the noun functions. They are also equivalent to 0 for the adjective functions in the V/A and V/N/A classes.

If some members of the V/N and V/N/A classes were limited, like John's and he, in their ability to take two or more determiners, the system would be unworkable because it would be impossible to code for verb transitivity and determiner limitation simultaneously, but since only genitive nouns, proper nouns, and pronouns have the determiner limitation and since they are always unambiguously nouns, no conflict arises. No word coded with a 1, 3, or 7 in first position has a 0, 4, or 5 in fourth position, with two exceptions: yes and no, which may be either nouns or sentences, are coded 3000.

To illustrate, the complete code for set is 3603. The first 3 indicates that it belongs to the class V/N; the 6 that it may be infinitive, present tense, past tense, or past participle; the 0 that as a present tense verb it requires a plural subject and as a noun it is singular;

the 3 in fourth position that as a verb it may take two objects.

Whenever the addition of an inflectional suffix resolves an ambiguity existing in the stem form of a word, the suffixed form is coded as unambiguously as possible. Sets, for example, is coded 3033 rather than 3633, since the s suffix resolves the ambiguity of the verb inflection, but not the overall V/N ambiguity.

Some suffixes create ambiguity. All present participles of verbs can function as nouns and some can also function as adjectives. Accordingly, setting is coded 3003 and exciting is 7002; cf., "the exciting of the atom," "it is exciting the atom," "it is very exciting." On the other hand, settings and findings are coded 2020 as unambiguous plural nouns capable of accepting determiners.

Many past participles of transitive verbs may premodify nouns, but they are not coded as potential adjectives unless they also accept the typical adjective modifiers more, most, very, and the inflection -ly. Broken, coded 5302, belongs to the latter class; consequently it was broken can be interpreted either as a passive or as a copulative construction, whereas it was taken cannot. Some words like underdeveloped or outspoken, although

morphologically marked as if they were past participles, are coded as adjectives only, since there are no verbs to underdevelop or to outspeak, and it was underdeveloped and he was outspoken are clearly copulative rather than passive.

The second position codes for the comparative and superlative forms of adjectives need little explanation. The -er suffix is a homograph and some words ending in it may belong to the N/A class. An agentive noun like farmer is coded 2000, but cleaner is coded 6700 as an ambiguous N/A-comparative. The superlative suffix is not homographic and its addition to an ambiguous stem resolves the ambiguity. Cleanest, clearest, and finest are all coded 4700, although clean and clear are 5002 and fine is 7003.

The -ly suffix, regarded as an adverbial inflection of adjectives, also resolves ambiguity and appears only with words having a 4 in first position. Words like deadly, kindly, and silly are coded as uninflected adjectives.

Using the code symbol 5 in third position for the -ly suffix, thus interrupting the numerical sequence of codes for noun subclasses in that position, is inadvertent and inelegant, but does no harm. It may also seem odd that the suffix was coded in this position rather than in second position. The reason for doing so is that only one digit

(9) was still available in second position and it was needed for coding the genitive nouns. Coding the genitives in second position avoids interference with the singular/plural coding in third position. It would be possible and perhaps preferable to make two symbols available in third position for genitives, one for the singular and one for the plural, and put the adjective suffix in second position. This may be done in a later revision.

All genitive nouns are ambiguously singular or plural with respect to subject-verb agreement, but not with respect to determiner-noun agreement; cf., "That man's accounts are false but this man's are true." Since genitives are unambiguously nouns, the code symbol 9 in second position will always be preceded by the code symbol 2 for words in this set.

The symbol 0 in third position codes words with no suffixes or with suffixes already coded in second position. The coding of the V/N suffix -s in third position has been explained earlier, and the use of the other symbols should be clear from the examples in Table 5.

The Determiners

The determiners are primarily subclassified according to two intersecting criteria: their ability to function as

substantives and their requirements for singular-plural agreement. These criteria establish four of the five subclasses coded in second position.¹ A single determiner, all, is arbitrarily assigned to a fifth subclass, because it is the typical outermost determiner in a noun phrase and isolating it simplifies writing the parsing rules. The classification is designed primarily to handle the determiners actually occurring in our rather limited glossary, but some additions have been anticipated in designing it. Both, for instance, which does not appear in our text, should probably be subclassified with all and then further differentiated.

Third position codes for subclassifying this group are designed differently from those for other classes and subclasses discussed so far. They are arranged in blocks so that as many as three different symbols may code the same syntactic function, thus permitting the assignment of unique codes to most of the determiners, and the writing of special parsing rules for their combinations without barring further writing of more general rules as well whenever possible. These blocks of digits subclassify the determiners according to their order classes, but only very roughly,

¹One theoretically possible subclass turns out to be empty, namely the "never substantive; plural agreement" subclass.

since the order class patterning of these words appears to be too intricate for the construction of any simple or elegant scheme. Codes 0, 1, and 2 are assigned to determiners which cannot follow any others. Codes 3, 4, and 5 are assigned to those which can follow the, this, that, those, his, her, etc., hereafter called the th- group, but which cannot follow all. Code 6 is assigned to numerical determiners three, four, etc., (but not one or two), which can follow both all and the th- group. Codes 7 and 8 are assigned to those which can follow all but not the th- group. Code 9 is assigned to such, which can follow any of the other determiners except a and the th- group.

Table 6 contains the codes and their meanings for second and third positions in the determiner group. Table 7 shows all of the determiners occurring in the glossary together with their codes, so that the reader can interpret the parsing rules for their individual combinations.

Later, other determiners can be inserted as the glossary is expanded and finer distinctions made, by using the fourth position for subclassification. Meanwhile, this design allows one to write parsing rules for determiner combinations such as many more, this much, some three ("approximately three"), which are always constitutes, as well as for anomalous combinations in which requirements for singular/

Table 6
Determiner Codes

Pos. 1	Position 2		Position 3		Pos. 4	
	Code	Code	Meaning	Code	Meaning	Code
8	0	0	Never substantive; singular agreement	0	Does not follow other determiners	0
	1	1	May be substantive; singular agreement	3	Can follow the <u>th</u> -group, but not <u>all</u>	
	2	2	May be substantive; plural agreement	4	Can follow the <u>th</u> -group and <u>all</u>	
	3	3	May be substantive; singular or plural agreement	5	Can follow <u>all</u> but not the <u>th</u> -group	
	4	4	Never substantive; singular or plural agreement	6	Can follow any determiner except <u>a</u> and the <u>th</u> -group	
	5	5	<u>All</u> : may be substantive; singular or plural agreement	7		

Table 7
Coded Determiners

80xx	81xx	82xx	83xx	84xx	85xx
8000 a	8100 much	8200 several	8300 some, any	8400 no	8500 all
8110 either	8210	8310 enough	8410		
8120 another	8220	8320	8420		
8130	8230 many	8330 more	8430		
8140 little	8240 few	8340 most	8440		
8150 one	8250 two	8350 other	8450		
8160	8260 three, four, etc.	8360 same	8460		
8170 this, that	8270 these, those	8370	8470 the		
8180	8280	8380 his, its, etc.	8480 my, your, her, etc.		
8190	8290	8390 such	8490		

plural agreement are altered, as in a little, a few, many a, and such a.

To, Be, Auxiliaries, and Contracted Forms

Subclassification of this group is illustrated in Table 8, where all of the words belonging to the group are provided codes whether or not they appear in our working glossary.

The infinitive marker to is assigned the unique code 9000. Since it is also a preposition/adverb, its code appears in the parsing rules for combining prepositions with objects, and verbs with adverbs. Although its range of syntactic functions is peculiar, there seems to be no need at this stage for assigning it two different word-numbers and two codes.

Like the intransitive verbs in the main word-classes, be and the auxiliaries all contain a 1 in fourth position, which becomes a 0 in the resultant code when they are parsed with potential preceding subjects. An indicative sentence with a form of be as its main verb will be coded 91x0 (with the third position containing a 0, 1, or 2). This means that copulative sentences are easily distinguishable from those containing other verbs. Likewise, elliptical or anaphoric sentences with forms of have, do or modals are

Table 8
To, Be, Auxiliaries, Contracted Forms

Pos. 1	Position 2	Code	Meaning	Code	Meaning	Position 3	Words	Code	Meaning	Position 4
9	0	Infini- tive marker	0	Infinitive, or plu- ral subject re- quired; or contrac- tions with <u>'ve</u>	to are, were, have, I've, we've, etc.			0	Infinitive marker	
1	Forms of <u>be</u>	1	Singular subject required; or con- tractions with <u>'s</u> (other than <u>let's</u>)	is, was, has, he's, etc.	does	1	Has no subject if <u>be</u> or auxili- ary; no verb or complement if contracted			
2	Forms of <u>have</u>	2	Singular or plural subject; or con- tractions with <u>'re</u>	had, did, can, we're, you're, they're						
3	Forms of <u>do</u>	3	Present participle; or contractions with <u>'d</u>	being, having he'd, they'd, etc.						
4	Modals	4	Past participle of <u>be</u> ; or contrac- tions with <u>'ll</u>	been he'll, they'll, etc.						
5	Contrac- ted forms	5	Infinitive of <u>be</u> ; or <u>let's</u>	be let's						
6		6	First person singu- lar form of <u>be</u> ; or contraction with <u>'m</u>	am I'm						

identifiable in the resultant codes, all of which will contain a 9 as first position code and 0 as fourth position, with the second position distinguishing the particular type.¹

In third position, codes 0, 1, and 2 are assigned on the basis of requirements for number agreement with subjects. Be is the only verb requiring this agreement in past tense as well as present tense forms, consequently the code 9121 is not assigned to any of its single forms. Modals, on the other hand, never require number agreement and are all coded 9421. When a modal, e.g., can, is parsed with the infinitive be, requirements for number agreement vanish and the code 9121 is assigned to the constitute. (Cf., he was, they were, he can be, they can be.)

Be differs from all other verbs also in having different forms for the infinitive and for the first-person singular, present tense. Codes for these forms are assigned arbitrarily in the third position.

Third position codes are also arbitrarily assigned to contracted forms of subject-plus-auxiliary and to let's. Contracted forms with n't are not coded separately, but are assigned the same word numbers and codes as the positive forms.

¹See Sec. IV, Resultant Codes.

Odd Forms, Numerals, Foreign Words

The remaining forms in the text, from which the working glossary comes, are not English words. They are numerals or alphabet sequences, not known to be acronyms, or foreign words like hoc, et, bleu. Hyphenated sequences of numerals which could be dates have been coded as nouns. The rest are assigned the first two position codes 96 and subclassified in third position. Their complete codes are:

Alphabetic only	9600
Mixed numerals and alphabet	9610
Arabic numerals	9620
Roman numerals	9630
Latin	9640
French	9650

Only three rules have been written for parsing them. One rule allows some of them to be parsed with nouns as if they were adjectives, and two rules provide for coordinating them with and. Such forms will have to be dealt with sooner or later--their frequency is higher than one might assume without checking--but other problems have priority at this point.

So far, the coding system has been presented primarily as it is used to code individual words and forms, although several times the extended application of word codes to

resultant codes for constitutes functioning like a single word has been touched on. A fuller discussion of the resultant codes, both those that correspond to word codes and those that appear only in the course of parsing, will be deferred until the parsing rules have been explained.

III. THE PARSING RULES

If we take the word as our smallest unit and the sentence as our largest, we can describe parsing as essentially the combining of words in a text into successively larger constituents until all of the words between two sentence-end markers have been accounted for. The codes attached to the words indicate what combinations they may enter; parsing rules provide the entry.

METHOD OF APPLICATION

Over 500 rules appear in Appendix B in a highly condensed form capable of expansion in the computer into an estimated 2,500. These rules simply state what adjacent grammar codes can be combined and what resultant codes their combination will produce, in the general form:

xxxx + yyyy → zzzz. The program will direct the computer to examine all adjacent words in the input sentence, compare their original codes (0_n and 0_{n+1}) with the pairs combined by the rules, apply all rules with matching pairs, and carry the resultant codes (R_x) in the continuing cycle. As the cycle continues, the pairs $0_{n-1} + R_x$, $R_x + 0_{n+2}$, and $R_x + R_{x+1}$ will also be checked against the rules.

For example, assume the input sentence is, "The men went there." The original grammar codes for each word will

be assigned in a glossary look-up subroutine. The codes, in order, are:

The (0_1)	men (0_2)	went (0_3)	there (0_4)
8470	2020	1101	0150

Comparing the adjacent codes with those in the rules, the computer assigns resultants as follows:

1. $0_1 + 0_2 \rightarrow R_1$: the/men
8470 2020 2024 determined noun
2. $0_2 + 0_3 \rightarrow R_2$: men/went
2020 1101 1000 sentence
3. $0_3 + 0_4 \rightarrow R_3$: went/there
1101 0150 1101 past tense, finite, intransitive
4. $0_1 + R_2 \nrightarrow$ no rule : * the/men went
8470 1000 (not permitted)
5. $R_1 + 0_3 \rightarrow R_4$: the men/went
2024 1101 1000 sentence
6. $R_2 + 0_4 \rightarrow R_5$: men went/there
1000 0150 1000 sentence
7. $R_1 + R_3 \rightarrow R_6$: the men/went there
2024 1101 1000
8. $0_1 + R_5 \nrightarrow$ no rule : * the/men went there
8470 1000 (not permitted)
9. $R_4 + 0_4 \nrightarrow$ no rule : * the men went/there
1000 0150 (not permitted)

The code 1000, which first appears in step 2, designates a constitute that is potentially a sentence or independent clause and is the resultant code whenever a finite verb or verb phrase, not requiring number agreement, is parsed with a possible subject preceding it. The appearance of this code in steps 2, 5, and 6 is premature because, although the resulting constitutes can be sentences, not all of the words in the string being parsed have been accounted for. Only step 7 accounts for all of the words in the string, and it alone will survive as the final resultant; the premature parsings will die in the cycle. The survivor will appear in an output consisting of steps 1, 3, and 7, corresponding to the completed parsing path.

A sample output for several test sentences is shown in Appendix C.

RULE SYMBOLS

The rules in the example are merely schematic. The actual parsing rules are written in a much more generalized and compact form for machine interpretation and manipulation, the compactness being achieved by using alphabetic cover symbols for various numerical code patterns, by establishing instruction symbols, and by writing rules as arrays, when more than one pair of codes produce the same resultants.

Cover Symbols

Suppose, for example, we wish to write a rule stating that a potential adjective plus a potential noun may be put together as a nominal constitute: $A + N \rightarrow N$. The number of different codes for words that could participate in this construction is extremely great; it will include not only the unambiguous codes for words like happy and men, but the ambiguous ones for words like man, deer, brief, set, abstract, cleaner, and exciting as well. In the system there are about sixty different codes for words that could be adjectives and about fifty for words that could be nouns. If every possible pair of codes were recorded separately, 3000 entries in the parsing rule for putting potential adjectives and nouns together would have to be rewritten.

Instead, cover symbols have been defined for each of the four code positions. Thus an A in first position covers the adjective code numbers 4, 5, 6, and 7;¹ an A in second position covers 0, 2, 3, 4, 5, 6, 7, and 8; an S in third position covers 0 and 4; and an S in fourth position covers 0, 1, 2, and 3 (indicating a simple unmodified adjective in a preceding A context). Similarly for nouns, N in first position covers the noun code numbers 2, 3, 6, and 7, and

¹See Table 1.

so forth for the other positions.¹ Now, the rule can be written:

AASS + NNNS → Resultant nominal constitute.

Instruction Symbols

The resultant code for the pair in the above example has not been specified because its third position code depends on whether the original noun constituent was singular or plural or potentially either. Since similar situations in writing other rules will be encountered, another kind of symbol, an instruction symbol, must be defined, which, in effect, orders the computer to copy or modify, in the resultant code, all or part of one of the constituent codes whenever the parsing rule is applied. Four such symbols have been defined.

B, in any position of the resultant code in a rule, means, "Copy in the resultant code for this pair, the code symbol in the corresponding position of the first constituent's code." C means, "Copy the code symbol in the corresponding position of the second constituent's code." The rule for combining adjective and noun now appears in the form:

AASS + NNNS → 20C0.

¹See Appendix A.

A related instruction symbol is D, which directs the computer to subtract 1 from whatever code digit appears in the corresponding position of the first constituent. Its main function is to allow one to write resultant codes for verbs or verb phrases plus their objects, setting the transitivity of the resulting constitute at one less than that of the original verb constituent. Thus, a verb that may take either one or two noun objects, represented with cover symbols by VVV2, will be parsed with a following noun 3 to produce a resultant in which the original 2 or 3 will be reduced by 1 and copied in the fourth position.

Presumably we could write the rule as:

 VVV2 + NNNN → 1BBD.
 3 6
 9N
 6

The addition of the 9 and 6 would be necessary since the symbol N in these positions does not cover the genitive and accusative noun codes. This is only a minor problem, as will become evident in a moment. More serious is the fact that use of the instruction symbol B in the resultant may lead to forbidden sequences. If the original verb was a V/N 303x or 304x, then a 103x or 104x would appear in the resultant. Such sequences should be avoided because they are inconsistent with the unambiguous 1 coding of the first

position. This can be done by splitting the rule into two rules of the following forms:

WS2 + NNNN \rightarrow 1B0D, where S = 0, 4¹
3 6
9N
6

and WV12 + NNNN \rightarrow 101D.

3 6
32 9N
3 6

In these forms of the rules, the ambiguities of the original codes are reduced as much as possible in the resultant.

These two rules may be simplified still further by using the fourth instruction symbol, an asterisk, which means, "Any code symbol is acceptable in this position"; in effect, an instruction to "skip it." In these two rules, it does not matter whether or not the verb is finite, or present or past participle; all that matters is that it is a verb and that it is transitive. Also, the following noun may be singular or plural, genitive or accusative. In other words, while the third position code of the potential verb must be specified in order to write the resultant, only the first and fourth positions are relevant for determining whether or not a given pair can be put together as verb plus

¹See Appendix A.

object. Therefore, we can write:

V*S2 N**N 1B0D
3

and V*12 N**N 101D.
3
32
3

Unlike the other instruction symbols B, C, and D, which appear only in resultants, the asterisk appears only in the constituent codes and never in the resultants.¹

RULE FORMAT

The rules are identified with an alphabetic symbol followed by a number. In keypunching, the fields used are:

<u>Columns</u>	<u>Contents</u>
1-2	Deck identification
5-8	Rule identification
15-18	Code of first constituent
25-28	Code of second constituent
35-38	Code of resultant

In order to interpret more fully the format in which the rules are written, let us return to our pair of verb-object

¹For a fuller understanding of how the instruction and cover symbols are used, the reader might try working his way through a few of the rules in Appendix B, after reading the next section in which the format is explained, and the key to Rule Symbols in Appendix A.

rules, noting that in the verb constituents some code positions were occupied in a second row. Thus, in the second rule of the pair, V^*12 was written. This is to be inter-

3
32
3

preted: "If the first position code is a V, the second position is not significant for application of this rule, the third position may be a 1 or a 3, and the fourth position may be a 2 or a 3." In general, whenever a code field for a constituent is not filled out to all four positions, each blank is interpreted as being filled by the symbol in the column above it and all combinations of four code position symbols are permitted. However, when all four positions are filled in both rows, as in $\begin{matrix} \text{aaaa} \\ \text{bbbb} \end{matrix}$, the interpretation is aaaa or bbbb.

We also interpret the connections across code fields as and/or connections, so that $\begin{matrix} \text{aaaa} \\ \text{bbbb} \end{matrix} + \begin{matrix} \text{cccc} \\ \text{dddd} \end{matrix} \rightarrow \begin{matrix} \text{eeee} \\ \text{ffff} \end{matrix}$ means: aaaa and bbbb or aaaa and cccc go to dddd or eeee. These conventions permit us to write rules as condensed arrays whenever more than one subclass or class of constituents can enter into a constitute, or whenever more than one subclass or class of constitutes are produced by the combination.

For example, the subclasses of transitive verbs may be parsed with some subclasses of determiners as well as with

nouns to produce resultant verb-object constitutes. The codes for such determiners may be added to previous rules. Accordingly, the second of the pair of verb-object rules, as it actually appears in the appendix, reads:

R26	V*12	N**N	101D
	3	8W*0	
	32		
	3		

(W is the cover symbol for the second position codes of all potentially substantive determiners). If the code field combinations were written out, leaving only the cover and instruction symbols intact, the rule would appear as

V*12	N**N	101D
V*13	N**N	101D
V*32	N**N	101D
V*33	N**N	101D
V*12	8W*0	101D
V*13	8W*0	101D
V*32	8W*0	101D
V*33	8W*0	101D

What its appearance would be with the cover and instruction symbols taken away, shall be left to the digital computer.

IV. RESULTANT CODES

TYPES

For purposes of discussion, the resultant codes are divided into three types on the basis of their resemblance to the constituents producing them and to the codes already defined for the glossary entries. Type A resembles the code of one of the constituents, indicating that the construction it codes is endocentric. For a highly endocentric construction, in which the syntactic range of one of the constituents is effectively the same as that of the constitute, the resultant code repeats the code of that constituent. In others, the resultant is a modified version of one of the constituent codes. The modified version will also be a glossary code, indicating that the combination of the two constituents has shifted the syntactic range of one of them to that of a different subclass of its own class. Occasionally, the Type A resultant combines subclass symbols in a way not found in the glossary, as in the combining of a modal with be to produce the code 9121, but no new subclass symbols are introduced and none of the previously defined symbols acquire new definitions.

Type B does not resemble the code of either constituent, indicating that the construction is exocentric. However, the

Type B resultant code is also a glossary code. This means that the syntactic range of the constitute is the same as that of a class of single words or forms not represented by its constituents.

Type C may code either endocentric or exocentric constructions, but it contains a code symbol not previously used for any subclass of glossary words or forms, or else not previously defined for a subclass of its own class. This indicates that the syntactic range of the constitute is a new one, appearing only when two or more glossary classes are combined.

In the following discussion, Type A and B resultants are briefly exemplified, but Type C is presented in more detail and the new code symbols and new uses for code symbols are listed and defined.

Type A resultants are the most frequent. Among others, they appear in the modifier-noun, the verb-object, the verb-adverb, the auxiliary-verb, and the determiner-determiner constitute codes. When simple adjectives are combined with nouns, for example, the noun code is repeated in the resultant. When a transitive verb is combined with an object, the resultant code is a modified version of the verb code, in which the transitivity has been reduced. Resultant codes for prepositional phrases and adverbial clauses also belong

to this type, both being coded as subclasses of adverbs, whose major class membership they share.

Type B resultants, though relatively infrequent, can be exemplified by the resultant code for marked infinitives (infinitives with to), which are coded as fully determined singular nouns (2xx5). Also, the 8W classes of determiners, the potential substantives, when combined with following prepositional phrases or postnominal adverbs, are coded as nouns.

Type C resultants are fairly frequent. They introduce one newly defined code symbol in second position, and one in third position, the rest appearing in fourth position. Table 9 shows them, together with the major classes in which they represent new subclassifications, and defines their meanings.

As shown in Table 9, the one newly defined code symbol (8) in second position is applied to nouns with post-modifiers. The rules provide that noun phrases will be put together in one order only, in order to reduce the number of final parsings. For instance, in the happy man on the corner, the head noun man will acquire its adjective, then the prepositional phrase, and then the determiner.

The only newly defined code symbol in third position is applied to relative clauses. Since these clauses are postnominal modifiers, as are prepositional phrases and some a'-verbs, they are coded in the first two positions as

Table 9
New Type C Resultant Code Symbols

Major Class	Pos.	Code	Meaning	Example
Noun	2	8	A post-modifier has been attached.	man on the corner
Adverbial	3	4	Postnominal modifier only; does not modify verbs.	which he wanted; whom I know; that came
Finite verbs, auxiliaries, contractions	4	0	Independent clause.	he came; let's go; he is, he has, he does; he can; he's here
Finite forms of <u>be</u>	4	2	Adverbial or adjective complement has been attached.	is happy; is on the corner
Finite forms of <u>be</u>	4	3	Inversion with <u>there</u> or <u>here</u> .	there is (a man here); here are (the men)
Adjective	4	4	"Determined" adjective.	very happy; very happily
Finite verbs, nouns	4	6	Interrogative clause, not inverted; or interrogative noun phrase, not potential introducer of relative clause.	who came; what books
Finite verbs, auxiliaries	4	7	Inverted interrogative clause, or interrogative noun phrase; may introduce a relative clause.	did he come; is he; has he, did he; can he; whose books

Table 9--Continued

Major Class	Pos.	Code	Meaning	Example
Finite verbs, auxiliaries	4	8	Doubly inverted interrogative clause.	when did he come; where is he; when can he; what books does he want
Any	4	9	A coordinating conjunction has been added; requires a parallel second constituent.	the man and (woman); coming and (going); happy and (gay); he came and (they left); can and (will); here and (there)

if they were adverbial; but since they do not modify verbs, the code symbol 4, not previously defined in third position for the preposition-adverb-conjunction group, is used to differentiate them from single word postnominal adverbs, prepositional phrases, and adverbial clauses with subordinating conjunctions. Accordingly, who is on the corner is coded 0340.¹ For example, compare:

the man who is on the corner	the man waited when he was on the corner
the man yonder	the man waited yonder
the man on the corner	the man waited on the corner

We quickly admit that calling relative clauses "adverbial" and then specifying that they do not modify verbs is an etymological contradiction, but will leave it to the reader to invent a more felicitous terminology if he wishes.

¹It is also coded as a noun phrase and as an interrogative clause. See the section on Multiple Resultants.

In fourth position, the code symbol 0 acquires a new definition when in the course of parsing it appears with preceding code symbols for finite verbs and auxiliaries. In the glossary, all potential verbs and auxiliaries have a 1, 2, or 3 in fourth position. When their preceding codes appear with a 0 in fourth position, it means that they have been combined with potential subjects and that the resultant constitute is an independent clause or sentence. In effect, this is to treat clauses as endocentric constructions with the verb or other finite form as head. Similarly, when a contracted form such as he's is combined with an appropriate second constituent such as coming, the resultant will be a modified version of the second constituent code, with a 0 in fourth position.

Two other fourth position symbols are redefined to provide for constructions containing a finite form of the verb be. When one of these forms is combined with a following adjective or adverbial expression, the fourth position of the resultant code is a 2, and the rules do not permit the addition of further adjective or noun complements. When one of the forms is combined with a preceding there or here, the fourth position code of the resultant is a 3, so that the common inversion patterns there is, here are, etc., can be identified and correctly parsed with following noun subjects.

The fourth position symbol 4 has previously been defined for nouns in the glossary to mean "determined subclass." No other use was specified for it in the glossary codes. In resultant codes, however, it too acquires a new definition. When adjectives are combined with a preceding modifier like very, the resultant code is a modified adjective code with a 4 in fourth position, meaning "determined adjective," i.e., it can no longer be parsed with words like more and most. The meaning is very close to that for the noun, but unlike the nouns, the adjectives as single words have no determined subclass--only some adjectival constituents do--and therefore we regard these resultants as Type C.

The completely new fourth position symbols 6, 7, and 8 code interrogative phrases and clauses. The symbol 6 means "interrogative, normal order" and is applied to questions like, "What books came yesterday?" (1106), and, "Who is here?" (9116), and to noun phrases like, "What books?" (2026).

The symbol 7, in fourth position, codes inverted clauses and appears in resultants for combinations of auxiliaries or forms of be with following substantives that may be their subjects. Is the man, accordingly, is coded 9117 and has the man, 9217. Subsequently, if is the man is combined with going, the resultant code is

1017; and if has the man is combined with gone, the resultant code is 1107. On the other hand, the rules provide that a subsequent combination, such as who with is the man will produce the resultant code 9116, indicating that the clause is a copulative interrogative, not inverted, and the combination of that with is the man will produce the resultant code 9110, indicating the clause is copulative, not interrogative, and not inverted.

The symbol 7 in fourth position is also used for noun phrases that may introduce relative clauses; for example, whose books (2027), which differs in syntactic range from what books (2026).

The symbol 8 in this position codes doubly inverted clauses like, "Where is the man?" and, "What books does he want?" in which an interrogative adverbial or object expression precedes the copulative verb or auxiliary, and the subject follows it.

The symbol 9 in fourth position codes combinations in which the first constituent is a coordinating conjunction. Strictly speaking, these are not constituents, but the rules establish them as temporary or pseudo constituents and provide that they may subsequently be parsed with preceding parallel expressions. For example, a singular or plural substantive expression combines with and to produce the resultant 2xx9. This in turn may combine with either a singular or plural substantive expression to produce a

plural noun phrase coded 2x20 (or 2x24 or 2x25, if the first noun belongs to a determined subclass).

The use of such temporary constituents can be extended to cover comparative constructions like a happier man than he, as happy as he, etc., as well as the either-or, neither-nor constructions, for which rules have not yet been developed.

MULTIPLE RESULTANTS

Although combining two constituents usually reduces ambiguity, in some instances ambiguity is created or increased and consequently more than one resultant code must be specified for the constituent. This is exemplified by clauses beginning with whose. The expression, whose books are here is potentially a question, a postnominal modifier, and a singular noun phrase. The rules for coding such expressions provide three resultants: 9106, 0340, and 2005. Similarly, who came is coded 1106, 0340, and 2005. Most of the multiple resultants appear in Rules U62-U72. As with words like do and have, which have more than one code, each code will be considered for possible combination with adjacent codes in executing the parsing program.

V. THE OUTPUT

The codes and rules have been tested, using a program written in IPL-V, a list-processing language. As was expected, the program proved excessively time consuming, requiring approximately ten minutes for parsing long sentences. A program currently being developed, using SCAT and adapted specifically to these codes and rules, has reduced the time to a matter of seconds.

A variety of sentences was hand coded, bypassing a dictionary look-up subroutine, and fed into the computer together with the rules and the program. The computer parsed them "correctly" (i.e., as predicted). Appendix C displays some of these sentences with their glossary and resultant codes and the rules which led to completed parsings, all represented in a tree structure form clearly showing the relationships of the substructures. Except for the summary information on sentence length and number of resultants and parsings, Appendix C is the actual output obtained automatically from an IBM 7090.

The output includes both very short sentences, designed to test the ability of the rules to handle basic sentence types, and longer sentences, taken from text with little modification, to test for the resolution of ambiguities. Specifically, the problem was to determine whether the codes and rules as developed so far could handle diverse structures commonly found in English and resolve the

ambiguities preserved in the coding of homographs sufficiently to avoid an excessive number of parsings. On both counts, the test results appear sufficiently favorable to warrant continuing development of this approach to automatic parsing.

Multiple parsings still pose serious problems. In general, syntactic ambiguities tend to compound themselves so that for each unresolved ambiguity in a sentence, one may expect that the number of parsings obtained for that sentence will be doubled. That is, one ambiguity in the subject phrase and one in the predicate phrase will produce four parsings, not two. On the other hand, this also means that each time codes or rules are added or refined to resolve a needlessly ambiguous construction, the number of different parsings for sentences containing the construction will probably be halved.

Most of the multiple parsings appearing for some of the test sentences were caused by the presence of prepositional phrases and other adverbial modifiers. The codes and rules are not now sufficiently refined to specify when these constituents should be attached to the sentence as a whole or to some one of several constituents within it. The addition of rules for punctuation will help, although the problem cannot be solved by this means alone. More accurate codes and rules for prepositional phrases would greatly reduce the number of unwarranted parsings.

One immediate step might well be the recoding of the preposition of and the writing of special rules for phrases in which it occurs. It is one of the most frequent words in ordinary text; it usually attaches its phrase to an immediately preceding noun, and the classes of verbs and adjectives to which an of phrase can be attached are limited. The most extreme example of multiple parsings in the test sentences was sentence 15, for which thirty parsings were obtained. The number would be halved if the phrase of text were attached solely and unambiguously to the preceding noun, passages.

As the codes and rules now stand, all sentences ending with two or more prepositional phrases or with a prepositional phrase following a noun object, will receive multiple parsings. Frequently, such sentences are truly ambiguous, as in the test sentence, "I saw the man with the telescope in the park," and all the parsings are legitimate interpretations. Sometimes they are not, as in, "The boy put the book on the table," and "The boy read a book on mathematics," where only one parsing is warranted. The problem of eliminating the unwarranted parsings while preserving legitimate ambiguities will obviously require finer coding of noun, verb, and adverbial classes.

An additional ambiguity is present in, "I saw the man with the telescope in the park," and twice as many parsings would have been obtained for it if the additional glossary

code for saw as a present tense form of the verb to saw had been read in. In a completely automated processing of text, this would occur. Its elimination here was entirely arbitrary and was done solely to save needless processing time rather than to minimize the gravity of the problem of homography. To prevent absurd interpretations in which men may be dissected with so blunt an instrument may require the use of microglossaries or the development of a classification scheme for semantic co-occurrence classes.

Similarly, the homography of a single word doubled the number of parsings in test sentence 14, where the sequence code combines subclass symbols occurs. In one set of parsings, combines is interpreted as a plural noun, modified by the singular noun code, and subclass is interpreted as the verb with symbols as its object. In the other set, code is interpreted as the subject, combines as the verb, and subclass as the noun modifier of the object noun symbols. Another unexpected ambiguity turned up in the test sentence, "Is the man master there." The answer to the first parsing presumably could be, "Yes, the man is master there"; to the second, "No, the man master isn't, but the woman master is." These examples serve to raise the question of the use of larger contexts in the resolution of ambiguities and suggest one of the dimensions in which the present limited scope of parsing is inadequate.

In spite of these problems, which would be difficult to solve in implementing any approach to automatic parsing, there are many advantages to the approach presented here. It preserves legitimate ambiguities. It does not try to force a narrow interpretation of "sentence" upon every string of words appearing between two end-marks of punctuation. We do not need to speculate or predict whether or not there will be a predicate after we have encountered a noun phrase. If there is one, and there is no subordinating context, the string of words will be parsed as an independent clause or sentence and its final resultant code will mark its type as indicative, copulative, anaphoric, or interrogative, or some combination of these, and its order as normal, inverted, or doubly inverted. If it is not, and the string is simply a noun phrase, as in a title, or some other so-called elliptical construction like the man over there or not until tomorrow, the final resultant code will so label it. Whether the total structure is a phrase, clause, or complex sentence, this approach permits its substructures and their connections to be clearly marked and the nature of the connections to be clearly indicated.

Appendix A

RULE SYMBOLS

Any position:	*	The code in this position is not significant for the application of this rule.
	B	Copy the digit in the corresponding position of the first constituent code.
	C	Copy the digit in the corresponding position of the second constituent code.
	D	Subtract 1 before copying the digit in the corresponding position of the first constituent code.
Position 1:	M	1,...7 Content word classes
	V	1,3,5,7 Verb
	N	2,3,6,7 Noun
	A	4,5,6,7 Adjective
Position 2:	V	0,...6 Verb
	N	0,2,5,6,7 Noun, not genitive
	A	0,2,3,4,5,6,7,8 Adjective
	S	0,5,6 Simple finite or infinitive
	T	1,4,6 Past tense verb
	F	0,1,4,5,6 Finite verb
	P	3,4,5,6 Past participle
	W	1,2,3,5 Noun substitute determiner
	X	2,5,6,7 Prep sition

	Y	3,5,7	Adverb
	Z	4,6,7	Subordinating conjunction
Position 3:	V	0,1,3,4	Verb
	N	0,2,3,4	Noun, not accusative
	A	0,4,5	Adjective
	S	0,4	Simple form
	Y	0,1,2,3	Adverb
Position 4:	V	1,2,3	Verb
	N	0,...5	Noun
	S	0,1,2,3	Unmodified noun or adjective

Appendix B

RULES

The rules are, in general, grouped according to the number of the resultant codes appearing in the right-hand column. The first column is the rule identification tag. The second column contains the codes of preceding constituents; the third column contains the codes of following constituents; the fourth column contains the resultant codes.

R1	N**N	V T*V	1000
	8W*0		
R2	N*4N	V S VV	1000
	83*0		
	5		
R3	N*0N	V*1V	1000
	81*0		
R4	N*2N	V S S V	1000
	3		
	82*0		
	9541		
	5		
R5	9511	V 2\$V	1000
	4	P	
R6	9521	V P\$V	1000
R7	9531	V S S V	1000
		P	
R8	4*5N	1**0	CCCC0
	0Y00	91*0	
	1	2	
	2	3	
	0150	4	
	6		
R9	20*6	1**7	1CC8
	7		
	0810		
	2		
	4		
	5		
	6		
	7		
R10	0110	1**7	1CC8
	2	91*7	
	3	2	
		3	
		4	
R15	9101	V 2*V	1088
	7	P	
	11		
	7		
R16	9121	V 2*V	1108
	7	P	
R17	9201	V P*V	1088
	7		
	11		
	7		
R18	9221	V P*V	1108
	7		
R19	9301	V S S V	1088
	7		
	11		
	7		

R20	9321 7 4 1 7	VSSV	1108
R25	V*52 3	N**N 8W*0	1800
R26	V*12 3 32 3	N**N 8W*0	1010
R27	V*SV	0YY0 405N 0150 6	1801
R28	V*1V 3	0YY0 405* 015* 6	1011
R29	405N 0Y10 2	V*SV	1000
R30	4050 0Y10 2	V*1V 3	1010
R35	A*SN NNS 29*0 VPS2 3	NNSS	2000
R36	A*SN NNS 29*0 VPS2 3	NN2S 3	2020
R38	NNSS	0300 2	2880
R39	NN2S 3 3	0300 2	2820
R40	8000 11 2 4	2850 4 9 0 4	2005
R41	8200 1 2 4	2820 4 920 4	2025
R42	8300 1 40 1 50	28*0 4 9 0 4	2005

R43	8130	2850	2004
	4	9	
	5		
	6		
	7		
	8		
	9		
R44	8230	2820	2024
	4	9	
	5		
	6		
	7		
	8		
R45	8330	2840	2004
	4	9	
	5		
	6		
	7		
	8		
	9		
	43		
	4		
	5		
	6		
	7		
	8		
R46	N*0S	VP*V	2005
	81*0		
R47	N*2S	VP*V	2025
	3		
	82*0		
R48	N*4S	VP*V	2045
	83*0		
	5		
R50	8000	NNSS	2005
	1	4	
	2		
	10		
	1		
	2		
R51	8200	NN2S	2025
	1	4	
	2	3S	
		4	
		4S	
		4	

R52	8300	NN2S	2025
	1	4	
	2	3S	
	40	4	
	1		
	2		
	50		
	1		
	2		
R53	8300	NNSS	20C5
	1	4	
	2		
	40		
	1		
	2		
	50		
	1		
	2		
R54	81*0	0340	2005
R55	82*0	0340	2025
R56	83*0	0340	2045
	5		
R57	8130	NNSS	2004
	4	4	
	5		
	6		
	7		
	8		
R58	8130	29SS	2904
	4		
	5		
	6		
	7		
	8		
R59	8230	NN2S	2024
	4	4	
	5	3S	
	6	4	
	7	4S	
	8	4	
R60	8330	NNSS	20C4
	4	4	
	5		
	6		
	7		
	8		
	9		
	43		
	4		
	5		
	6		
	7		

		8	
		9	
		53	
		4	
		5	
		6	
		7	
		8	
		9	
R61	8330	NN2S	2024
	4	4	
	5	35	
	6	4	
	7		
	8		
	9		
	43		
	4		
	5		
	6		
	7		
	8		
	9		
	53		
	4		
	5		
	6		
	7		
	8		
	9		
R62	9131	N**N	2004
		8W*0	
		AASS	
		0Y*0	
		0150	
		6	
		VP*V	
R63	28*0	0340	2005
	9		
R64	NNSS	0340	2005
R65	NN2S	0340	2025
	3		
R66	81*0	0300	2800
		2	
R67	82*0	0300	2820
		2	
R68	83*0	0300	2840
	5	2	
R69	90*0	VSSV	2005
		9151	
		3	
		5	
R70	96*0	N**S	2000

R71	N**N	9600 1 2 3	2000
R75	0810 2 6	NNSN	2006
R76	0840	NNSN	2007
R77	0810 2 7	NN2N	2026
R78	0840	NN2N 3	2027
R79	0810 2	29*N	2906
R80	0060 9	ASAS 2	4004
R81	0100	VP*V	
R82	0100	470S 4	4705
R83	4050	ASSS 2	4004
R84	VP*V		
R85	8330	ASSS 2	4700
R86	VP*2 3		
R87	8330	A*5S	4050 8
R88	8340	ASAS 2	4800
R89	VP*2 3		
R90	ASAS 2	8310	4005
U0	8000	8140 2	8300
U1	8170	8100 823*	8000
U2	8170 4	8150	8150
U3	8100 3 4 5 6 20	8330	8880
U4	8300 4	8330	8300

U5	8300 4	8140 5 6 24	8C10
U6	8300 4	8350	8120
U7	8370 8 47 8	8340 5 6	8370
U8	8500	8260 7	8210
U9	0060	8100 4 8230 4	8000
U10	0110	8100	0860
U11	0110	8230	0870
U12	8230 8390	8000	8000
U20	9101 7 11 7	9131	9188
U21	9201	9141	9101 5
U22	9211 7 21 7 31	9141	9188
U23	9207	9141	9107
U24	94*1 7	9151	9128
U25	0040	9131	90C8
U26	9W01 7 11 7 21 7	0040	9B88
U30	9511	N*SN 81*0 3 5 A*SN VP*2 3	9100

U31	9511	0150	9100
		6	
		0Y*0	
U32	N**N	9121	9120
	8W*0	2	
		7	
U33	N*SN	9111	9110
	81*0	2	
	3	7	
	5		
U34	N*2N	9101	9100
	3	2	
	4	7	
	82*0		
	3		
	5		
U35	91*1	0YY0	91B2
	2	0150	
		6	
		A**N	
U36	91*0	0YY0	91B8
	6	0150	
		6	
		405N	
U37	91*1	0150	91B3
U38	0150	91*7	91C0
	6		
U39	0Y00	91*6	91CC
	1	8	
	2		
	405N		
U40	N**N	9221	9C20
	8W*0	3	
		4	
U41	N*SN	9211	9C10
	81*0	3	
	3		
	5		
U42	N*2N	9201	9C00
	3	3	
	4		
	82*0		
	3		
	5		
U50	9101	N*2N	9B07
	2	3	
	3	4	
	9103	82*0	
		3	
		5	
U51	9111	NSSN	9B17
	2	8	

	2	8	
	3	9	
	9113	81*0	
		3	
		5	
U52	9121	N**N	9827
	2	8W*0	
	3		
	4		
U53	91*7	0150	9180
	8	6	
		0YY0	
		A**N	
U54	9107	N*2N	9108
		3	
		4	
		82*0	
		3	
		5	
U55	9117	N*SN	9118
		81*0	
		3	
		5	
U56	9127	N**N	9128
		8W*0	
U59	9151	N**N	9155
		A*SN	
U60	0800	VFVV	0340
		91*1	
		7	
		8	
		92*1	
		3	
		4	
U61	0810	1000	2005
	2*6	91*0	
		2	
		3	
		4	
U62	9231	VP*V	0340
			2005
U63	080*	1000	0340
	2	91*0	2005
	5	2	
	2**7	3	
		4	
U64	0840	1000	0340
			2005
U65	0810	VFVV	1006
	2046		2005
U66	0860	VTVV	1000
	7		2005
	2**6		

U67	0860	V\$IV	1006
	2006	3	2005
U68	0870	VSSV	1006
	2026		2005
U70	0820	VFVV	1006
	3		0340
	2047		2005
U71	2007	V\$IV	1006
		3	0340
			2005
U72	2027	VSSV	1006
			0340
			2005
U73	0X*0	N**N	0300
	9000	8W*0	
U80	0810	9101	91C6
	2*46	7	2005
		11	
		7	
		21	
		7	
U81	2*06	9111	91C6
	0860	7	2005
		21	
		7	
U82	2*26	9101	91C6
	0870	7	2005
		21	
		7	
U83	0820	9101	91C6
	3	7	0340
	4	11	2005
	2*47	7	
		21	
		7	
U84	2007	9111	91C6
		7	0340
		21	2005
		7	
U85	2027	9101	91C6
		7	0340
		21	2006
		7	
U90	0030	0040	0010
U91	0010	M***	CCC9
	2	8W**	
	3	961*	
		2	
		01**	
		A	
		910*	
		1	
		2	
		942*	

U99	VF*V	VF*9	188C
Q0	N**N	N**9	2025
	8W*N	8W*9	
Q1	VF*0	VF*9	188U
Q2	V2*V	V2*9	1201
Q3	VP*V	VP*9	1301
Q4	AVSS	AVS9	4000
Q5	4*50	4059	4050
Q6	A7*S	A7*9	4700
Q7	4800	4809	480U
Q8	91*0	91*9	9188
	1	2	
		3	
		4	
Q9	96*0	96*9	968U
Q10	0110	0119	018U
	2	2	
	3	3	
Q11	0X*0	0X*9	0239
Q12	0YY0	0YY9	CCCCU
	0150	0159	
	6	6	
Q13	0340	0349	0340
Q20	N**N	0190	8888
	A**N		
	8W*N		
	0Y*0		
Q25	0040	NNNN	CCCC5
		AAAA	
		8W*0	
		0150	
		6	
		0Y*0	
Q27	0050	M**N	CCCC
		8W*0	
		0150	
		6	
		0Y*0	
Q28	NN*N	005*	CCCC
	9		
	8W*0		
Q30	0110	1**0	0330
	2	91*0	
	3	2	
	0Z*0	3	
		4	

Appendix C

SAMPLES OF PARSED SENTENCES

Most tree-structure representations of sentences show nodes with branches to the left and right, according to the order in which the elements appear in the sentence. The trees appearing on the following pages, however, branch to the right and down. The code appearing in the upper left-hand corner is to be considered as the topmost node. It is as if a normal tree had been converted to its mirror image and rotated through an angle of 45°.

The rules by which each node was constructed appear beneath the resultant code. (It should be noted that a 0 in the first position of a code was eliminated in the print-out.)

The number of different parsings obtained for each sentence is recorded and for some sentences, all the parsings are printed out. Where there were very many parsings, only a few are printed out, since the others are simply alternate combinations of the ambiguous elements in the ones already displayed. In addition, the number of resultants appearing during the processing is recorded. Some of these resultants do not survive, but the number affords a better index to the length of time required for processing than does the length of the sentence.

SENTENCE 1

Sentence: Let's try to parse this one.
Length: 6 words
Number of parsings: 1
Number of resultants: 13

1000 **** 9541
R4 LET'S
*
*
1001 **** 3002
R25 TRY
*
*
2005 **** 9000
R69 TC
*
*
1001 **** 1002
R25 PARSE
*
*
8150 **** 8170
U2 THIS
*
*
8150
ONE.

SENTENCE 2

Sentence: We hope that this will run.
Length: 6 words
Number of parsings: 1
Number of resultants: 19

1CCC **** 2C25
R4 WE
*
*
1CC1 **** 3CC2
R25 HCPE
*
*
2CC5 **** C8CC
U63 THAT
*
*
1CCC **** 8170
R1 THIS
*
*
1C1 **** 9421
R2C WILL
*
*
3CC2
RLN.

SENTENCE 3

Sentence: Some men were coming and going.
Length: 6 words
Number of parsings: 1
Number of resultants: 11

1CCC **** 2C25 **** 8300
R4 R52 SCME
*
*
* 2C20
* MEN
*
*
1CC1 **** 91C1
R15 WHERE
*
*
12C1 **** 1209 **** 1201
Q2 U91 COMING
*
*
* CCI0
* ANC
*
*
12C1
COMING

SENTENCE 4

Sentence: Why did he go?
Length: 4 words
Number of parsings: 1
Number of resultants: 6

1108 **** C130
R1C WHY
*
*
1107 **** 9327 **** 9321
R2C U52 CID
*
*
*
* 2005
* HE
*
*
1001
GC

SENTENCE 5

Sentence: There is a man there.
Length: 5 words
Number of parsings: 1
Number of resultants: 6

911C **** 9110 **** 0150
U36 U38 THERE
*
*
*
* 9117 **** 9111
* U51 IS
*
*
* 2C05 **** 8000
* R50 A
*
*
* 3C02
* MAN
*
*
*
015C
THERE

SENTENCE 6

Sentence : Is the man there?
Length : 4 words
Number of parsings : 1
Number of resultants : 4

9118 **** 9117 **** 9111
U53 L51 IS
* *
* 2004 **** 8470
* R60 THE
* *
* 3002
* MAN
*
*
0150
THERE

SENTENCE 7

Sentence: Is there a man there?
Length: 5 words
Number of parsings: 1
Number of resultants: 6

9118	****	9117	****	9113	****	9111
053		051		037		IS
*		*		*		
*		*		*		
*		*		0150		
*		*		THERE		
*		*		*		
*		*		*		
*		2005	****	8000		
*		R50		A		
*		*		*		
*		*		*		
*		3002				
*		PAR				
*						
*						
015C						
THERE						

SENTENCE 8

Sentence: The man is happy there.
Length: 5 words
Number of parsings: 2
Number of resultants: 11

9110 **** 2004 **** 8470
U33 R60 THE
*
*
* 3002
* MAN
*
*
* 9112 **** 9112 **** 9111
U35 U35 IS
*
*
* 4000
* HAPPY
*
*
* 0150
THERE

9110	****	9110	****	2004	****	8470
U36		U33		R60		THE
*		*		*		
*		*		*		
*		*		3002		
*		*		MAN		
*		*				
*		*				
*		9112	****	9111		
*		U35		IS		
*		*				
*		*				
*		4000				
*		HAPPY				
*						
0150						
THEIR						

SENTENCE 9

Sentence : Is the man happy there?
Length : 5 words
Number of parsings : 1
Number of resultants : 4

9118 **** 9118 **** 9117 **** 9111
U53 U53 U51 IS
* * *
* * * 2004 **** 8470
* * R60 THE
* * *
* * * 3002
* * MAN
* *
* * 4000
* * HAPPY
* *
* *
0150
THERE

SENTENCE 10

Sentence: Is the man master there?
Length: 5 words
Number of parsings: 2
Number of resultants: 11

9118	****	9117	****	9111
U53		U51		IS
*		*		
*		*		
*		2004	****	8470
*		R60		THE
*		*		
*		*		
*		2000	****	3002
*		R35		MAN
*		*		
*		*		
*		3002		
*		MASTER		
*		*		
*		*		
0150				
THERE				

SENTENCE 11

Sentence: Who is in charge of production there?
Length: 7 words
Number of parsings: 2
Number of resultants: 21

9116	****	9116	****	9116	****	0830
U36		U36		U83		WHO
*		*		*		
*		*		*		
*		*		9111		
*		*		IS		
*		*		*		
*		*		*		
*		0300	****	0530		
*		U73		IN		
*		*		*		
*		*		*		
*		2800	****	3002		
*		R38		CHARGE		
*		*		*		
*		*		*		
*		0300	****	0230		
*		U73		OF		
*		*		*		
*		*		*		
*		2000				
*		PRODUCTION				
*						
*						
0150						
THERE						

9116	****	9116	****	9116	****	9116	****	0830
L36		L36		U36		U83		WHO
*		*		*		*		
*		*		*		*		
*		*		*		9111		
*		*		*		IS		
*		*		*				
*		*		*				
*		*		0300	****	0530		
*		*		U73		IN		
*		*		*				
*		*		*				
*		*		3002				
*		*		CHARGE				
*		*						
*		*						
*		0300	****	0230				
*		U73		OF				
*		*						
*		*						
*		2000						
*		PRODUCTION						
*								
*								
015C								
THERE								

SENTENCE 12

Sentence: The boy put the book on the table.
Length: 8 words
Number of parsings: 2
Number of resultants: 27

1000	***	2004	***	8470
R1		R6C		THE
*		*		
*		*		
*		2000		
*		ECY		
*		*		
*		*		
1601	***	1601	***	1602
R27		R25		PUT
*		*		
*		*		
*		2004	***	8470
*		R6C		THE
*		*		
*		*		
*		3002		
*		ECCK		
*		*		
0300	***	0530		
U73		CA		
*		*		
*		*		
2004	***	8470		
R6C		THE		
*		*		
*		*		
3002				
TABLE.				

1CCC	****	2CC4	****	8470
R1		R60		THE
*		*		
*		*		
*		2CCC		
*		BCY		
*				
*				
16C1	***	16C2		
R25		PUT		
*				
*				
28C4	***	8470		
R45		THE		
*				
*				
28CC	***	3CC2		
R38		BLOCK		
*				
*				
03CC	***	0530		
U73		CA		
*				
*				
2CC4	***	8470		
R60		THE		
*				
*				
3CC2				
TABLE.				

SENTENCE 13

Sentence: I saw the man with the telescope in
the park.
Length: 10 words
Number of parsings: 4
Number of resultants: 37

1000 **** 2025
R1 I
*
*
1101 **** 1102
R25 SAW
*
*
2804 **** 8470
R45 THE
*
*
2800 **** 3002
R38 MAN
*
*
0300 **** 0230
U73 WITH
*
*
2804 **** 8470
R45 THE
*
*
2800 **** 3002
R38 TELESCOPE
*
*
0300 **** 0530
U73 IN
*
*
2004 **** 8470
R60 THE
*
*
3002
PARK

1000 **** 2025
R1 I
*
*
1101 **** 1101 **** 1102
R27 R25 SAK
* *
* *
* 2004 **** 8470
R60 THE
* *
* *
* 3002
* MAN
*
*
0300 **** C230
U73 WITH
*
*
2804 **** 8470
R45 THE
*
*
2800 **** 3002
R38 TELESCOPE
*
*
0300 **** C530
U73 IN
*
*
2004 **** 8470
R60 THE
*
*
3002
PARK

1CCC **** 2C25
R1 I

*
*
11C1 **** 11C1 **** 1102
R27 R25 SAW

*
*
*
* 28C4 **** 8470
R45 THE

*
*
* 28C0 **** 3002
R38 MAN

*
*
* C300 **** 0230
U73 KITH

*
*
* 2C04 **** 8470
R6C THE

*
*
* 3CC2
TELESCOPE

*
*
* 03C0 **** C530
U73 IN

*
*
* 2C04 **** 8470
R6C THE

*
*
* 3CC2
PARK

1000 **** 2025
R1 I
*
*
1101 **** 1101 **** 1101 **** 1102
R27 R27 R25 SAW
* * *
* * *
* * * 2004 **** 8470
* * * R60 THE
* * *
* * *
* * * 3002
* * * MAN
* * *
* * * 0300 **** 0230
* * * U73 WITH
* * *
* * *
* * * 2004 **** 8470
* * * R60 THE
* * *
* * *
* * * 3002
* * * TELESCOPE
* * *
* * *
0300 **** 0530
U73 IN
*
*
2004 **** 8470
R60 THE
*
*
3002
PARK

SENTENCE 14

Sentence: Occasionally the modified code
combines subclass symbols in a
new way but no new undefined symbols
are introduced.

Length: 18 words

Number of parsings: 10* (Five parsings only shown.)

Number of resultants: 249

*Seven of the ten parsings are due to coding the
word combines 3032.

1000 **** 4050
R8 OCCASIONALLY
*
*
1000 **** 1000 **** 2004 **** 8470
C1 091 R3 R60 THE
* * * *
* * * * 2000 **** 1402
* * * * R35 MODIFIED
* * * * *
* * * * 3002
* * * * CODE
* * * *
* * * * 1011 **** 3032
* * * * R26 COMBINES
* * * *
* * * * 2820 **** 3002
* * * * R37 SUBCLASS
* * * *
* * * * 2820 **** 2020
* * * * R39 SYMBOLS
* * * *
* * * * 0300 **** 0530
* * * * U73 IN
* * * *
* * * * 2005 **** 8000
* * * * R50 A
* * * *
* * * * 2000 **** 4000
* * * * R35 NEW
* * * *
* * * * 2000
* * * * WAY
* * * *
* * * * CC30
* * * * BUT
* * * *
* * * * 1000 **** 2025 **** 8400
R4 R52 NC
* * * *

* 2020 **** 4000
* R36 NEW
*
* 2020 **** 4000
* R36 UNDEFINED
*
* 2020
* SYMBOLS
*
* 1001 **** 9101
R15 ARE
*
* 1402
INTRODUCED.

1000 **** 4050
R8 CCCASIONALLY
*
*
1000 **** 1000 **** 2004 **** 8470
Q1 U91 R3 R60 THE
* * * * *
* * * * * 2000 **** 1402
* * * * * R35 MODIFIED
* * * * * *
* * * * * 3002
* * * * * CODE
* * * * *
* * * * * 1011 **** 3032
* * * * * R26 COMBINES
* * * * *
* * * * * 2820 **** 2020 **** 3002
* * * * * R39 R36 SUBCLASS
* * * * *
* * * * * 2C20
* * * * * SYMBOLS
* * * * *
* * * * * 0300 **** 0530
* * * * * U73 IN
* * * * *
* * * * * 2005 **** 8000
* * * * * R50 A
* * * * *
* * * * * 2000 **** 4000
* * * * * R35 NEW
* * * * *
* * * * * 2000
* * * * * WAY
* * * * *
* * * * * CC30
* * * * * BLT
* * * * *
* * * * * 1000 **** 2025 **** 8400
R4 R52 NC
* * * * *

* 2C20 **** 4000
* R36 NEW
*
*
* 2C20 **** 4000
* R36 UNDEFINED
*
*
* 2C20
* SYMCLS
*
*
1001 **** 9101
R15 ARE
*
*
1402
INTRODUCED.

1000 **** 4050
R8 CCCASIONALLY

1000 **** 1009 **** 1000 **** 2004 **** 8470
Q1 U91 R3 R60 THE

1011 **** 1011 **** 3032
R28 R26 COMBINES

2020 **** 3002
R36 SUBCLASS

2020
SYMBOLS

0300 **** 0530
U73 IN

2005 **** 8000
R50 A

2000 **** 4000
R35 NEW

2000
WAY

0030
BLT

1000 **** 2025 **** 8400
R4 R52 NO

* 2020 **** 4000
* R36 NEW
*
*
* 2020 **** 4000
* R36 UNDEFINED
*
*
* 2020
* SYMBOLS
*
*
* 1001 **** 9101
* R15 ARE
*
*
* 1402
INTRODUCED.

* 2C20 **** 4000
* R36 NEW
*
*
* 2C20 **** 4000
* R36 UNDEFINED
*
*
* 2C2C
* SYMBOLS
*
*
1CCL **** 91C1
R15 ARE
*
*
14C2
INTRODUCED.

1000 **** 1009 **** 1000 **** 4050
Q1 U91 R8 OCCASIONALLY
* * * * *
* * * * * 1000 **** 2004 **** 8470
* * * * R3 R60 THE
* * * * *
* * * * * 2000 **** 1402
* * * * R35 MODIFIED
* * * * *
* * * * * 3002
* * * * * CODE
* * * * *
* * * * * 1011 **** 3032
* * * * R26 COMBINES
* * * * *
* * * * * 2820 **** 2C20 **** 3002
* * * * R39 R36 SUBCLASS
* * * * *
* * * * * 2C20
* * * * * SYMBOLS
* * * * *
* * * * * 0300 **** 0530
* * * * U73 IN
* * * * *
* * * * * 2005 **** 8000
* * * * R50 A
* * * * *
* * * * * 2000 **** 4000
* * * * R35 NEW
* * * * *
* * * * * 2000
* * * * * WAY
* * * * *
* * * * * 0030
* * * * * BLT
* * * * *
* * * * *
1000 **** 2025 **** 8400
R4 R52 NO
* * * * *
* * * * *
* * * * * 2C20 **** 4000
* * * * R36 NEW

* *
* *
* 2C20 **** 4000
* R36 UNDEFINED
* *
* *
* 2C20
* SYMBOLS
* *
* 1CC1 **** 91C1
* R15 ARE
* *
* 14C2
* INTRODUCED.

SENTENCE 15

Sentence: This paper describes a method for discriminating between passages of text received from different sources.

Length: 15 words

Number of parsings: 30 (Four parsings only shown.)

Number of resultants: 264

*
*
2C20
SOURCES.

1000 **** 2004 **** 8170
R3 R57 THIS
*
*
* 3002
* PAPER
*
*
1011 **** 1011 **** 1011 **** 1011 **** 1012
R28 R28 R28 R26 DESCRIBES
*
*
*
* 2005 **** 8000
* R50 A
*
*
* 2000
*
* MÉTHOD
*
*
* 0300 **** 0630
* U73 FOR
*
*
* 3201
* DISCRIMINATING
*
*
* 0300 **** 0230
* U73 BETWEEN
*
*
* 2020 **** 2020
* R39 PASSAGES
*
*
* 0230 **** 0230
* U73 CF
*
*
* 2000 **** 2000
* R46 TEXT
*
*
* 1402
* RECEIVED
*
*
* 0230 **** 0230
* U73 FROM
*
*
* 4000 **** 4000
* R36 DIFFERENT

•
•
2020
SOURCES.

1CCC **** 2CC4 **** 8170
R3 R57 THIS
*
*
* 3CC2
* PAPER
*
*
1C11 **** 1C11 **** 1011 **** 1011 **** 1012
R28 R28 R28 R26 DESCRIBES
*
*
*
*
* 2005 **** 8000
* R5C A
*
*
*
*
* 2000
*
*
*
*
*
*
* 0300 **** 0630
* U73 FCR
*
*
*
* 3201
* DISCRIMINATING
*
*
* C300 **** 0230
* U73 BETWEEN
*
*
* 2C25 **** 2820 **** 2020
* R47 R39 PASSAGES
*
*
*
* 0300 **** 0230
* U73 OF
*
*
*
*
* 2000
* TEXT
*
*
* 14C2
* RECEIVED
*
*
* 0300 **** 0230
* U73 FROM
*
*
* 2C20 **** 4CC0
* R36 CIFFERENT

2020
SOURCES.

1000 **** 2004 **** 8170
R3 R57 THIS
* *
* *
* 3002
* PAPER
* *
*
1011 **** 1012
R26 DESCRIBES
* *
*
2805 **** 8000
R4C A
* *
*
2800 **** 2000
R38 METHOD
* *
*
0300 **** 0630
U73 FCR
* *
*
2800 **** 3201
R38 DISCRIMINATING
* *
*
0300 **** 0230
U73 BETWEEN
* *
*
2820 **** 2020
R36 PASSAGES
* *
*
0300 **** 0230
U73 CF
* *
*
2005 **** 2000
R46 TEXT
* *
*
1401 **** 1402
R27 RECEIVED
* *
*
0300 **** 0230
U73 FROM
* *
*
2020 **** 4000
R36 DIFFERENT

*
*
2C2C
SOURCES.

SENTENCE 16

Sentence: These blocks of digits subclassify the determiners very roughly since the order-class patterning of these words appears to be too intricate for the construction of any simple or elegant scheme.

Length: 30 words

Number of parsings: 5

Number of resultants: 368

1000 **** 2824 **** 8270
R4 R44 THÈSE
*
*
* 2820 **** 3032
* R39 BLOCKS
*
*
* 0300 **** 0230
* U73 UF
*
*
* 2020
* CICITS
*
*
1001 **** 1001 **** 1001 **** 1001 **** 1001 **** 1002
R27 R27 R27 R27 R25 SUBCLASSIFY
*
*
*
*
* 2024 **** 8470
* R61 THE
*
*
*
* 2020
* DETERMINERS
*
*
* 4054 **** CC60
* R90 VERY
*
*
* 4050
* ROUGHLY
*
*
* 0330 **** 0700
* Q30 SINCE
*
*
* 1000 **** 2804 **** 8470
* R3 R45 THE
*
*
* 2800 **** 2000 **** 3002
* R38 R35 ORDER-CLASS
*
*
* 3202
* PATTERNING
*
*
* 0300 **** C230
* U73 UF

* * * * * 2024 **** 8270
* * * * * R59 THESE
* * * * *
* * * * * 2020
* * * * * WORDS
* * * * *
* * * * * 1011 **** 1012
* * * * * R26 APPEARS
* * * * *
* * * * * 2005 **** 9000
* * * * * R69 TO
* * * * *
* * * * * 9155 **** 9151
* * * * * U59 BE
* * * * *
* * * * * 4004 **** 0090
* * * * * R90 TCU
* * * * *
* * * * * 4000
* * * * * INTRICATE
* * * * *
* * * * * 0300 **** 0630
* * * * * U73 FCR
* * * * *
* * * * * 2004 **** 8470
* * * * * R6C THE
* * * * *
* * * * * 2000
* * * * * CONSTRUCTION
* * * * *
* * * * * 0300 **** 0230
* * * * * U73 LF
* * * * *
* * * * * 2005 **** 8300
* * * * * R53 ANY
* * * * *
* * * * * 2000 **** 4000 **** 4009 **** 4000
* * * * * R35 C4 U91 SIMPLE
* * * * *
* * * * *

0020
OR

4000
ELEGANT

3002
SCHEME

1CCC **** 2824 **** 8270
R4 R44 THESE
*
*
* 2820 **** 3032
* R39 BLOCKS
*
*
* C300 **** 0230
* U73 OF
*
*
* 2C20
* DIGITS
*
*
1CC1 **** 1CC1 **** 10C1 **** 1001 **** 1002
R27 R27 R27 R25 SUBCLASSIFY
*
*
*
*
* 2C24 **** 8470
* R61 THE
*
*
*
* 2C20
* DETERMINERS
*
*
*
* 4054 **** 0C60
* R90 VERY
*
*
* 4050
* ROUGHLY
*
*
*
* 0330 **** 0700
* C30 SINCE
*
*
* 1000 **** 2804 **** 8470
* R3 R45 THE
*
*
* 2800 **** 2000 **** 3002
* R38 R35 ORDER-CLASS
*
*
*
* 3202
* PATTERNING
*
*
*
* 0300 **** 0230
* U73 OF

2024 **** 8270
R59 THESE

2020
WORCE

1011 **** 1011 **** 1012
R28 R26 APPARES

2005 **** 9000
R69 TG

9155 **** 9151
U59 BE

4004 **** 0090
R90 TCG

4000
INTRICATE

0300 **** 0630
U73 FOR

2004 **** 8470
R60 THE

2000
CONSTRUCTION

0300 **** C230
U73 CF

2005 **** 8300
R53 ANY

2000 **** 4000 **** 4009 **** 4000
R35 C4 U91 SIMPLE

0020

CR

4000
ELEGANT

3002
SCHEME

1000 **** 2824 **** 8270
R4 R44 THESE
*
*
* 2820 **** 3032
* R39 BLOCKS
*
*
* 0300 **** 0230
* U73 OF
*
*
* 2020
* DIGITS
*
*
1001 **** 1001 **** 1001 **** 1001 **** 1002
R27 R27 R27 R25 SUBCLASSIFY
*
*
*
*
* 2024 **** 8470
* R61 THE
*
*
*
* 2020
* DETERMINERS
*
*
*
* 4054 **** 0060
* R9C VERY
*
*
* 4050
* RUGGELY
*
*
* 0330 **** 0700
* C30 SINCE
*
*
* 1000 **** 2804 **** 8470
* R3 R45 THE
*
*
* 2800 **** 2000 **** 3002
* R38 R35 ORDER-CLASS
*
*
*
* 3202
* PATTERNING
*
*
* 0300 **** 0230
* U73 OF

* * * * *
* * * * * 2024 **** 8270
* * * * * R59 THESE
* * * * *
* * * * * 2020
* * * * * WORDS
* * * * *
* * * * * 1011 **** 1012
* * * * * R26 APPEARS
* * * * *
* * * * * 2005 **** 9000
* * * * * R69 TC
* * * * *
* * * * * 9155 **** 9151
* * * * * L59 DE
* * * * *
* * * * * 4004 **** 0090
* * * * * R90 100
* * * * *
* * * * * 4000
* * * * * INTRICATE
* * * * *
* * * * * 0300 **** C630
* * * * * U73 FCR
* * * * *
* * * * * 2004 **** 8470
* * * * * R45 THE
* * * * *
* * * * * 2000 **** 2000
* * * * * R38 CONSTRUCTION
* * * * *
* * * * * 0300 **** C230
* * * * * U73 OF
* * * * *
* * * * * 2005 **** 8300
* * * * * R53 ANY
* * * * *
* * * * * 2000 **** 4000 **** 4009 **** 4000
* * * * * R35 C4 U91 SIMPLE
* * * * *
* * * * *

0020
CR

4000
ELEGANT

3002
SCHEME

1000 **** 2824 **** 8270
R4 R44 THESE
*
*
* 2820 **** 3032
* R39 BLOCKS
*
*
* 0300 **** 0230
* U73 OF
*
*
* 2020
* DICTS
*
*
1001 **** 1001 **** 1001 **** 1002
R27 R27 R25 SUBCLASSIFY
*
*
*
* 2024 **** 8470
* R61 THE
*
*
*
* 2020
* DETERMINERS
*
*
* 4054 **** 0060
* R9C VERY
*
*
* 4050
* RELICELY
*
*
0330 **** 0700
U30 SINCE
*
*
1000 **** 2804 **** 8470
R3 R45 THE
*
*
* 2800 **** 2000 **** 3002
* R38 R35 ORDER-CLASS
*
*
*
* 3202
* PATTERNING
*
*
* 0300 **** 0230
* U73 OF

* *
* *
* 2024 * * * 8270
* R59 THE SF
* *
* *
* 2C2C
* WCRDS
* *
* 1011 * * * 1011 * * * 1012
* R28 R26 APPEARS
* *
* *
* 2CC5 * * * 9000
* R69 TO
* *
* *
* 9155 * * * 9151
* U59 BE
* *
* *
* 4004 * * * 0090
* R5C TOO
* *
* *
* 4000
* INTRICATE
* *
* *
* 0300 * * * 0630
* U73 FCR
* *
* *
* 28C4 * * * 8470
* R45 THE
* *
* *
* 2800 * * * 2CC0
* R38 CONSTRUCTION
* *
* *
* 0300 * * * 0230
* U73 OF
* *
* *
* 2CC5 * * * 8300
* R53 ANY
* *
* *
* 2CCC * * * 4CCC * * * 4009 * * * 4000
* R35 C4 U91 SIMPLE
* *
* *
* *

OC20
UR

4CC0
ÉLÉGANT

3CC2
SCHEME

1000 **** 2824 **** 8270
R4 R44 THESE
* *
* *
* 2820 **** 3032
* R39 BLOCKS
* *
* *
* C300 **** 0230
* U73 CF
* *
* *
* 2020
* DIGITS
* *
* *
1001 **** 1001 **** 1002
R27 R27 R25 SURCLASSIFY
* *
* *
* *
* * 2024 **** 8470
* R61 THE
* *
* *
* *
* * 2020
* DETERMINERS
* *
* *
* * 4054 **** 0060
* R90 VERY
* *
* *
* * 4050
* ROUGHLY
* *
* *
0330 **** 0700
Q30 SINCE
* *
* *
1000 **** 2804 **** 8470
R3 R45 THE
* *
* *
* * 2800 **** 3002
* R38 R35 ORDER-CLASS
* *
* *
* * 3202
* * PATTERNING
* *
* *
* * C300 **** 0230
* U73 CF

2024 **** 8270
R59 THESE

2C2C
KCRCS

1011 **** 1011 **** 1011 **** 1012
R28 R28 R26 APPEARS

2005 **** 9000
R69 TC

9155 **** 9151
U59 BE

4004 **** 0090
R90 TCO

4000
INTRICATE

C300 **** 0630
U73 FCR

2004 **** 8470
R60 THE

2000
CONSTRUCTION

0300 **** C230
U73 CF

2005 **** 8300
R53 ANY

2000 **** 4000 **** 4009 **** 4000
R35 C4 U91 SIMPLE

0020
CR

4000
ELEGANT

3CC2
SCHEME